Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes

Andreas Reinhardt
School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
andreasr@cse.unsw.edu.au

Daniel Burgstahler
Multimedia Communications Lab
Technische Universität Darmstadt
Darmstadt, Germany
daniel.burgstahler@kom.tu-darmstadt.de

***Abstract*—Platform heterogeneity in wireless sensor networks is often seen as a major challenge for application development. Once embedded systems with different processor architectures, computational power, and memory are part of the same network, algorithms and applications must be adapted to this additional degree of complexity. As a result, current sensor network deployments are (with exception of the sink node) commonly comprised of devices of identical make and model. In this paper, we show how device heterogeneity may be exploited to improve the energy efficiency of the sensor network by shifting resource-intensive processing tasks to other nodes within the network. To this end, we analyze the energy demand for representative processing operations and wireless communications on six heterogeneous state-of-the-art sensor platform types. Based on the created models, we assess the achievable energy savings when tasks are shifted to more powerful processing nodes. Our results show that platform heterogeneity, although often being perceived as a hindrance to the easy deployment of applications, also serves as an enabler for increased energy efficiency of the network.**

I. Introduction

Wireless Sensor Networks (*WSN*s) serve the purpose to collect, process, and wirelessly communicate readings collected from the physical environment to a data sink [1]. The data collecting embedded sensing systems in WSNs (*motes*) are commonly based on identical hardware platforms, mainly to simplify application development and allow for an easy replacement of defective nodes. Only the sink node, which generally collects all sampled readings, is often assumed to be unconstrained in its energy budget and computational power. An example for a homogeneous topology is shown in the left part of Fig. 1, where all data collected by the leftmost node are communicated to the sink over a multi-hop wireless connection.

In current WSN research, however, many different mote platforms with heterogeneous capabilities co-exist. An even larger variety of heterogeneous devices is envisioned in the emerging Internet of Things [2]. A such heterogeneous WSN setting is depicted in the right part of the figure, showing two mote types with different computational power. The main objective of this paper is to exploit this device heterogeneity in order to process data within the WSN and therewith reduce its overall energy demand. This goal is achieved by migrating resource-intensive computation tasks to nodes that can execute them in the most energy-efficient way.

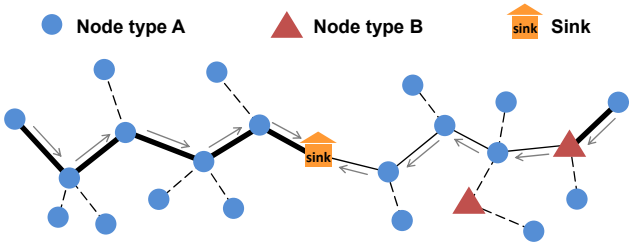


Figure 1. Homogeneous (left) and heterogeneous (right) WSN topology. The line width serves as an indicator for the volume of the transferred data.

Let us assume that nodes of type A in Fig. 1 are incapable of performing computationally demanding algorithms (e.g., data compression or cryptographic operations) locally due to limitations in their processing power and RAM. They can hence only forward collected sensor readings in unprocessed form, leading to a large volume of data flowing towards the sink; this is also visualized by the large width of the connecting line in the left part of the figure. If we now assume that nodes of type B provide sufficient computational power to process the data, the node of type B in the right part of the figure can be utilized to process the large data volume generated by the rightmost type A sensor. As a result, a significantly smaller volume of data needs to be transported to the sink. The usage of this data processing node is however only energetically feasible if the transmission of the resulting smaller data packets to the sink saves more energy than the amount required for the processing at node B.

In this paper, we investigate if and under which conditions a migration of processing tasks between computationally heterogeneous motes is feasible. Therefore, we determine characteristics of a set of six representative mote platforms and assess their performance by benchmarking their computational power. In addition, high-resolution electrical power measurements have been conducted in order to determine the energy demand for the benchmark on each platform. Based on the created models, we present an approach to assess if and when the migration of data processing tasks

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

to other nodes is feasible. Besides solely focusing on the data processing energy demand, our concept also takes all incurred communication costs into account. Finally, by using two representative evaluation scenarios, we prove that the presence of computationally heterogeneous devices can lead to an extended lifetime of the WSN.

We present these contributions as follows. In a first step, we summarize work related to heterogeneity in WSNs in Sec. II. We subsequently present an overview of our task migration concept and highlight its components in Sec. III. Based on the mote platforms used in current research, we study the heterogeneity of current motes and benchmark their power consumption characteristics to determine relevant model parameters in Sec. IV. We evaluate the performance and feasibility of our approach in Sec. V and conclude this paper in Sec. VI, where we also highlight future research directions.

II. Related Work

Heterogeneity in WSNs has been approached from several directions of research. Most of them, however, consider the unequal distribution of mote resources as an additional complication. The use of computationally heterogeneous devices, which are intentionally deployed to organize the WSN in a clustered structure, represents the only exception to this rule. Instead of featuring only a single sink node, these topologies are composed of a two-tiered architecture based on a large number of data collecting motes (i.e., cluster members) and a smaller number of cluster heads. In general, these cluster heads are assumed to be less constrained in terms of both their energy budget and their computational capabilities, and thus act as distributed sink nodes to different parts of the WSN. Often, separate communication links and protocols are employed to communicate the collected sensor data from the cluster heads to the sink node.

The addition of cluster heads equipped with wired backhaul links has been presented in [3]. These nodes are not confined in their energy budget and establish links to their cluster head neighbors and the sink. The tethered nature of these newly introduced cluster heads, however, enables them to communicate with each other directly, effectively turning the WSN into a network with multiple sink nodes. Wireless backhaul links between cluster heads have been proposed [4], [5], [6]. These solutions commonly rely on the use of a secondary communication channel, e.g., IEEE 802.11, to transfer data between the cluster heads. Yet, similar to the case of a tethered backhaul connection, the use of wireless communication between the cluster heads can be effectively considered to create a multi-sink WSN, which does not rely on the existing WSN infrastructure to transport data between the cluster heads. In fact, the cluster heads are rather considered as distributed data collection nodes instead of being integral parts of the WSN.

Similarly, the use of functional device heterogeneity has been proposed to increase the network lifetime in [7]. In the approach, two different kinds of devices are being used, namely so called *Endpoints* and *Routers*. While the former category of nodes only becomes active when data has been locally collected and needs to be transmitted, each Router also serves as a relay node for data and thus operates at higher duty cycles. Achievable energy savings are, however, not analyzed in the paper. In a later work, Yu et al. have shown a mechanism to optimize the number and locations of processing nodes with higher processing capabilities [8]. However, their approach also focuses only on the reduction of the energy consumption of the regular sensor nodes, not taking the overhead introduced by the newly placed processing nodes into account.

Besides using computationally powerful and energetically unconstrained nodes to add data sinks to the network, heterogeneity in WSNs also manifests itself in other forms. Heterogeneity in terms of link qualities or energy levels is, e.g., analyzed in [9], [10], although these domains can be considered orthogonal to the computational heterogeneity on which this paper focuses. The MacroLab programming framework is introduced in [11] and targets to abstract from platform-specific programming by instead defining a WSN's operation globally. The approach is thus specifically designed to overcome device heterogeneity by automatically decomposing and adapting applications to the WSN. In summary, however, these approaches target to mitigate device heterogeneity rather than exploit it.

Apart from configuring computationally powerful nodes as cluster heads, it can be observed that device heterogeneity is currently commonly seen as a hindrance to simple application development. We, however, argue that heterogeneity should also be seen as an opportunity to extend a network's operational time by reducing the volume of wireless traffic.

III. Task Migration in WSNs

We have observed that only few approaches exist that utilize computationally heterogeneous nodes in WSN research. In such approaches, the more powerful nodes are mainly used as dedicated data collecting devices with secondary backhaul links interconnecting each other. With the rise of the Internet of Things [2], however, an unprecedented number of heterogeneous *Smart Objects* can be anticipated to coexist in the same sensor network. Each of these systems will be designed to perform specific sensing or actuation tasks, and can thus be expected to be heterogeneous in different aspects, including the available memory and computational power. Nodes with different amounts of available resources will thus comprise an integral part of future WSNs, although their co-existence does not necessarily imply their co-operation towards the same goal, e.g., the maximization of the network's lifetime.

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

Table I
CROSS-SECTION OF CURRENT MOTE PLATFORM SPECIFICATIONS

Device	MCU	Word Size	Clock
Imote 2 [12]	Intel PXA271	32 bit	104 MHz
INGA [13]	ATmega 1284p	8 bit	8 MHz
Mulle v5.2 [14]	Renesas M16C/62P	16 bit	10 MHz
SunSPOT v6 [15]	AT91SAM9G20	32 bit	400 MHz
TelosB [16]	TI MSP430F1611	16 bit	4 MHz
XM1000 [17]	TI MSP430F2618	16 bit	8 MHz

In our novel task migration scheme, which we introduce in the remainder of this section, we prove the usefulness of shifting processing tasks between computationally heterogeneous nodes. It effectively exploits the heterogeneity in platform-specific characteristics like processor speeds and available memory sizes introduced by the variety of underlying mote platforms. By shifting resource-demanding tasks from low-power embedded systems to other motes which can handle the processing at a lesser energy overhead, overall energy savings are achieved. At the same time, reducing the volume of wireless traffic caters to the mitigation of channel congestion, which is known to represent a performance bottleneck in dense deployments with high data volume.

A. Preliminary Feasibility Study

The fundamental idea behind our scheme is to exploit the different hardware specifications of the microcontroller units (*MCU*s) used on current mote platforms. For reference, we have listed the MCUs and their properties of six current mote platforms in Table I. From the table, it becomes clear that clock frequencies differ significantly by up to a factor of a hundred, and that different MCU word sizes are being used across the devices. As a result, it can be assumed (note that we confirm these assumptions by experimental evaluations in Sec. IV) that the execution of a task on a computationally more powerful platform can lead to significant increases in their execution speed. At the same time, however, the power consumption of the MCUs clocked at higher frequencies often range significantly higher than the values of low-power embedded systems.

Let us give an example. The Mulle node's MCU requires 8mA in active mode [14], whereas the processor of the Imote 2 consumes 66mA [12], i.e., about 8 times as much. However, the Imote 2 is clocked at 104MHz, whereas the standard configuration of the Mulle node configures the internal system clock to 10MHz, such that the Imote 2 effectively runs 10 times as fast as the Mulle. The quotient of both values (8-fold increase in current consumption vs. 10-fold increase in computational speed, i.e., a factor of 0.8) serves as an indicator whether another platform may be capable of executing a task in a more energy-efficient manner. In this example, the quotient is smaller than 1.0, meaning that the Imote 2 can be expected to perform the processing more efficiently than the Mulle in the first approximation. It needs to be remarked at this point that even further savings can be expected, as effects like MCUs with complex instruction sets or pipelining architectures are not covered in this simple model. Yet, it serves its purpose well to convey the general idea behind our task migration scheme.

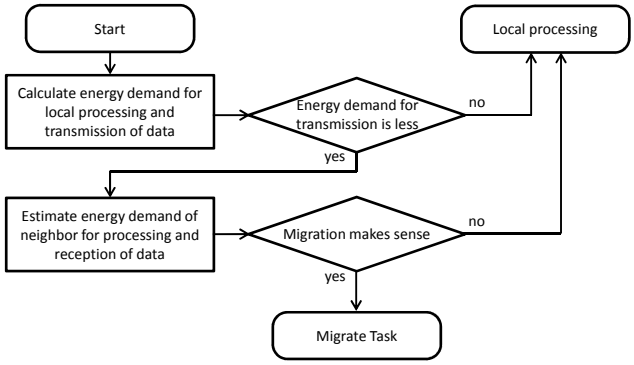


Figure 2. Typical operation flow of our system

B. General Concept

We have motivated that situations may exist in which nodes in the WSN might be capable of performing processing tasks in a more energy-efficient fashion. The logical next step is to analyze if and to which destination node tasks should be migrated to reduce the energy demand for their execution. As the global reduction of energy consumption in the network represents the main objective of our work, we analyze each step (i.e., processing, communication, etc) with regard to its required energy budget. Without loss of generality, let us therefore assume a collection-based WSN, e.g., utilizing the Collection Tree Protocol [18]. A routing tree is established, which is rooted at the sink, and all data are recursively forwarded until they reach this destination. Beacon packets are periodically broadcast by each node in order to facilitate the creation of the routing tree.

Our concept for the application to be executed on each data collecting sensor node is visualized in the flow diagram shown in Fig. 2, and can be explained as follows. As soon as a new sensor reading has been collected, the node locally estimates the energy demand required to process the data based on previous processing operations. In case no historical data is available, the processing step is executed once and the corresponding energy demand is established by multiplying the platform's power consumption (cf. Sec. III-C) with the execution duration of the task. Once the local energy consumption for the data processing is known, a two-step migration assessment process is triggered. At first, the node checks whether the wireless transmission of the data to a processing node would consume more energy than the local processing. In this case, the processing is done locally in order to preserve energy. Otherwise, if the local processing is known to exceed the communication cost, the system estimates the energy demands incurred by reception and

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

processing the data on each potential processing node, i.e., each one-hop neighbor (see Sec. III-D). If the sum of communication and remote processing energy demands is still less than what is required for the local processing, the migration process is triggered[1].

C. Computational Power Index

Prior to migrating tasks to other nodes in the network, an estimation of their energy demand to perform the processing task must be made. We have thus borrowed from the energy estimation approach in [19], in which execution times for methods are multiplied with pre-determined power consumption measurements for different operation modes. The approach chosen in the mentioned paper is to assume that the MCU's power consumption is static across all types of operations. Although this assumption is sufficient to explain the generic operation of our algorithm, we refine the analysis of node power consumptions in Sec. IV, especially as this linearized model cannot distinguish between RAM-intensive tasks and ones which mostly rely on register operations. Still, the product of the measured execution time with the power consumption value is being used to provide an estimation of the consumed energy. The benefit of this approach is that the execution time can be easily measured by any mote's internal timers. For the computational power index, we use the typical operating power P_{MCU} when the MCU is active.

D. Task Distribution

Although the distribution of tasks to arbitrarily located nodes in the network would be possible, we confine our task shifting to a node's direct neighborhood. This also mitigates the impact of stale information about node locations and renders specific point-to-point routing protocols unnecessary. Furthermore, confining the selection of a processing node to the local neighborhood avoids large deviations from the original route (i.e., the route determined by the given collection protocol). As a result, we only consider one-hop neighbors for the processing of data in this paper, which piggyback information about their computational power onto the regularly broadcast CTP beacons.

As mentioned before, we measure this computational power by means of the MCU's average power consumption in active operation mode, to which we refer as P_{MCU} and which is individual for each type of mote. In order to take the transmission costs into account, a second value P_{RxTx} is introduced that describes the average power demand for sending or receiving data. Even though the energy demand for sending and receiving differs slightly and the values also differ between different radio transceiver devices, we assume a static value in this paper to simplify our analyses. Since during transmissions the MCU is also active, the energy demand of packet reception (E_{Rx}) and transmission (E_{Tx}) is the sum of P_{RxTx} and P_{MCU}, multiplied with the transmission duration $t_{transfer}$ (cf. Eq. 1). As a result, each mote can also estimate the transmission costs of its neighbors, because the P_{MCU} value of the neighbors is known and P_{RxTx} is assumed to be constant for the WSN.

$$E_{Rx} = E_{Tx} = t_{transfer} \cdot (P_{RxTx} + P_{MCU}) \quad (1)$$

The condition to migrate processing to a neighboring node is shown in Eq. 2, in which all contributing elements to the energy balance are shown. In the equation, the indices l are used for the local provider of sensor data and p refers to the potential remote processing node. Note that the local transmission cost $E_{Tx,l}$ is not part of the equation because it is required in any case.

$$E_{compute,p} + E_{Rx,p} < E_{compute,l} \quad (2)$$

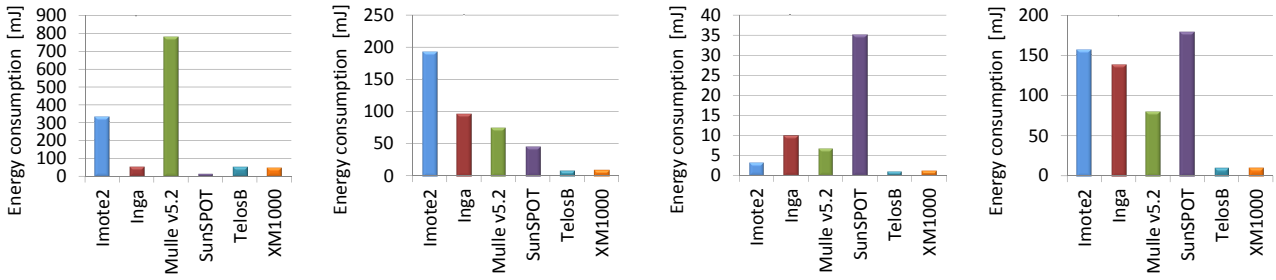
The only remaining unknown parameter in the set of equations is the transmission duration $t_{transfer}$. Through experimentation, we have determined a value of 5.96ms for $t_{transfer}$ for the transmission of an IEEE 802.15.4 packet of maximum size (i.e., a payload size of 114 bytes [20]) over an ideal and lossless communication channel. This includes both the time required to transfer the outbound data from the MCU to the radio transceiver's buffer as well as the actual wireless transmission. In case other packet sizes are being used, we approximate the resulting transfer duration using a linear function for simplicity reasons.

IV. Node Benchmarking

In Table I, we have shown that even a small cross-section of current mote platforms shows a strong heterogeneity in terms of their processing units. Considering the values specified in the corresponding MCU data sheets, their current consumptions during active operation mode also differ significantly, ranging from 1.8mA for the TelosB to 66mA for the Imote 2. Instead of relying on the averaged values provided in the MCU data sheet, however, we have decided to practically determine the overall power consumption of the six chosen mote platforms to get accurate values for the validation of our concept. A second reason to conduct practical measurements was the fact that further platform-specific components besides the MCU are present on the mote platforms. Whilst most of their data sheets only specify extremal current consumption values, our practical measurements include their typical current draw during regular operation. We specifically target the investigation of the actual values of P_{MCU} for the selected platforms. To this end, we make use of a benchmark, which is executed on the mote while their input voltage levels and current consumptions are measured simultaneously. Several aspects of benchmarking have been investigated for WSNs, e.g., [21], [22], [23], yet the unavailability of the proposed benchmarking suites has motivated us to implement a set of computationally demanding functions ourselves.

[1]For the sake of simplicity, we assume that each node is capable of performing computations on each data packet according to its required processing step (e.g., data compression, filtering, transforms, etc).

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.



(a) Fast Fourier Transform (b) LZSS compression (c) Twofish encryption (d) Fibonacci number calculation

Figure 3. Energy demand for the execution of four benchmarking functions on each platform

A. Benchmark Functions

Benchmark functions are necessary to determine the processing powers of the motes and put them into relation to their energy demand. We have specifically derived computationally demanding functions from realistic WSN tasks, which we discuss as follows.

1) Lossless Data Compression: The first considered task is data compression, which caters to the reduction of packet sizes by eliminating redundancies from radio packets prior to their transmission. Even if the amount of data gathered by each node is small, processing nodes can aggregate the data from several motes and compress it before forwarding it to the sink. We have implemented two data compression algorithms for WSNs, namely the Lempel-Ziv-Storer-Szymanski (LZSS) algorithm, a variant of LZ77, as well as the Lempel-Ziv-Markov-Algorithm (LZMA).

2) Data Encryption: The second task in our benchmark suite of computationally demanding algorithms is data encryption. Again, we have implemented two encryption algorithms, the first one being Blowfish, a symmetric key block cipher, and the second one being Twofish. An advantage of both algorithms is that pre-computed key-dependent tables are used. This has the benefit, especially for a battery powered mote in a WSN, that not all of the computing has to be done each time data has to be encrypted. However, for a better comparability, the key-dependent table computations are also taken into account for the measurements made in this paper.

3) Signal Processing: The third considered application task is high data rate signal processing, which also has big potential for energy savings. For some applications like seismic sensing or acceleration measurements, a sampling rate of tens to thousands readings per second is required. A typical processing operation on such high data rate samples is the analysis in the frequency domain, to which the signal can be converted by applying the Fast Fourier Transform (FFT). We thus complete our benchmarking suite with a split-radix algorithm to calculate the FFT of real numbers as well as a second FFT algorithm that operates on complex values.

4) Recursions: In addition to the previously mentioned algorithms, we have implemented a recursive algorithm to compute Fibonacci numbers to get a comparison of the behavior of branches in the program flow and the arithmetic computing capabilities of the different mote platforms.

B. Benchmark Function Parametrization

We have executed the previously introduced algorithms to get comparable values of the motes energy demand in relation to the processing power. For Fibonacci number calculation, we executed the algorithm with a pre-defined upper value of 25 recursion steps. The FFT functions were each executed for an input sample set of 1024 values, for which we have used acceleration readings collected from a mobile sensor worn on a person's wrist. As input for the compression and encryption algorithms, we have selected a set of environmental measurement data collected in Orly, France [24]. The data set contains samples of temperature, barometric pressure, and humidity, sampled in an interval of 30 minutes. We have executed the data compression algorithms with multiple subsets of the data of 1024 bytes each, and used subsets of 2048 bytes as input to the encryption algorithms.

C. Benchmark Results

We have run all of the presented benchmark functions on the selected mote platforms and measured the execution time and energy demand. Since measuring the real energy demand of sensor nodes can be difficult because of the low current consumptions in the range of milliamperes, we have used a Hitex PowerScale with ACM probe [25], capable of measuring currents from 200nA to 500mA. Both current and voltage are measured synchronously at a sampling rate of 100kHz. The results of our measurements are given in Table II, which shows the average current consumption as well as the execution duration for the complete benchmarking suite. The total energy demand to execute the benchmark is shown in the rightmost column, where differences by a factor of more than ten can be observed. For a visual comparison, the energy demands for the execution of four functions are furthermore depicted in Fig. 3.

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

Table II
MEASURED DURATION AND ENERGY DEMAND FOR THE BENCHMARK SUITE EXECUTED ON EACH CONSIDERED PLATFORM

Device	$\overline{P}$ *[mW]*	t *[s]*	E *[mJ]*
Imote 2	317.69	2.768	879.30
INGA	90.24	3.987	359.79
Mulle v5.2	39.75	30.086	1226.63
SunSPOT v6	385.61	0.830	320.18
TelosB	5.57	21.231	118.26
XM1000	11.21	10.299	115.46

Upon closer inspection of the data collected using the Hitex PowerScale, we have observed that only very slight changes to the MCU's power consumption occur depending on the nature of the performed operation. In fact, even when each of the MCUs was configured to solely execute `NOP` instructions, a change of less than 1μW could be observed across all considered platforms. The averaged $\overline{P}$, as listed in Table II, can thus be considered to be fully representative of each MCU's power demand, and we rely on these practically established values in the remainder of this paper.

V. EVALUATION

In order to evaluate the proposed task shifting concept, we have conducted extensive simulations based on the real-world measurements collected in our benchmark. In this section, we thus present the feasible node combinations for task shifting as well as analytically calculated energy savings for heterogeneous WSNs based on different network topologies.

A. Feasible Node Combinations

In a first step, we have analyzed which combinations of source and destination motes are suitable candidates for the migration of tasks. We have thus selected the Fast Fourier Transform as example task, with an assumed 1,024 bytes of input data that require $t_{transfer} = 47.71ms$ for their transmission over the wireless channel. Based on the average power consumption values determined in Sec. IV, we have determined which combinations of source and destination mote type satisfy the feasibility condition (i.e., Eq. 2 in Sec. III). The resulting energy demands for communication ($E_{Rx,p}$ and $E_{Tx,l}$) and computation ($E_{compute,p}$) as well as their sum are given in Table III. The heterogeneity

Table III
MEASURED ENERGY DEMAND FOR RECEIVING 1,024 BYTES OF DATA AND COMPUTING THE FFT OVER THIS DATA ON DIFFERENT NODES

Device	$E_{Rx,p}$ = $E_{Tx,l}$	$E_{compute,p}$	$E_{compute,p}$ $+E_{Rx,p}$
Imote 2	23.603mJ	329.41mJ	353.013mJ
INGA	9.275mJ	53.108mJ	62.383J
Mulle v5.2	6.095mJ	782.690mJ	788.785mJ
SunSPOT v6	27.881mJ	11.660mJ	39.541mJ
TelosB	3.941mJ	49.450mJ	53.391mJ
XM1000	4.297mJ	48.570mJ	52.867mJ

Table IV
PLATFORM COMBINATIONS WHERE TASK SHIFTING IS FEASIBLE

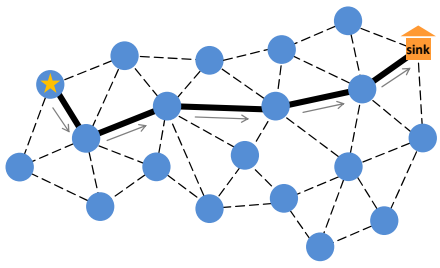
	Processing Node					
Source Mote	Imote 2	INGA	Mulle v5.2	SunSPOT v6	TelosB	XM1000
Imote 2	-	x	-	x	x	x
INGA	-	-	-	x	-	-
Mulle v5.2	x	x	-	x	x	x
SunSPOT v6	-	-	-	-	-	-
TelosB	-	-	-	x	-	-
XM1000	-	-	-	x	-	-

becomes very clear from the table; even when the additional communication costs are considered, the Mulle v5.2 device requires more than 20 times as much energy as the SunSPOT v6 mote to perform the FFT computation.

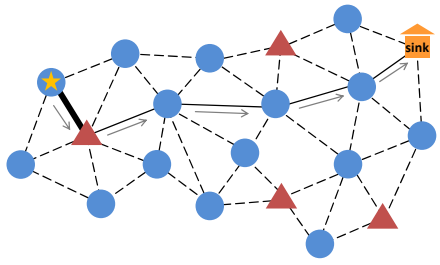
The resulting reasonable combinations of motes, i.e., constellations where the feasibility condition is fulfilled, are tabulated in Table IV. Mote platforms named on the left are the nodes considered as start nodes on which the sensor data is collected, while the column headers indicate the possible processing platform type. Each combination in which the feasibility condition from Eq. 2 is fulfilled is marked by a cross. The table shows clearly that the SunSPOT mote is well suited as processing node for data collected by all other source node types, whereas tasks to be performed at the Mulle mote should always be shifted to other node types for increased energy efficiency.

B. Analysis from a Local Perspective

In order to assess the practical use of our proposed task shifting concept, we have conducted analytical assessments of two WSN topologies. Since no simulator exists that



(a) Scenario A



(b) Scenario B

Figure 4. Topologies of the simulated scenarios

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

Table V
ENERGY DEMAND FOR DATA TRANSMISSIONS IN SCENARIO A (HOMOGENEOUS NETWORK WITHOUT ANY TASK MIGRATION)

Mote	$E_{Rx} = E_{Tx}$	$E_{total,A}$
Imote 2	23.603mJ	212,423mJ
INGA	9.275mJ	83,475mJ
Mulle v5.2	6.095mJ	54,851mJ
SunSPOT v6	27.881mJ	250,929mJ
TelosB	3.941mJ	35,473mJ
XM1000	4.297mJ	38,670mJ

Table VI
ENERGY DEMAND FOR INITIAL/SUBSEQUENT TRANSMISSIONS AND FOR THE EXECUTION OF THE COMPLETE PROCESSING TASK IN SCENARIO B

Mote	$E_{Tx,start}$	$E_{Rx,Tx,intermediate}$	$E_{total,B}$
Imote 2	23.603mJ	1.180mJ	71.618mJ
INGA	9.275mJ	0.464mJ	52.992mJ
Mulle v5.2	6.095mJ	0.305mJ	48.858mJ
SunSPOT v6	27.881mJ	1.394mJ	77.180mJ
TelosB	3.941mJ	0.197mJ	46.059mJ
XM1000	4.297mJ	0.215mJ	46.521mJ

natively supports energy models of all the selected motes, we have chosen to assess the energy consumptions for each scenario analytically. Again, we do not consider any specific MAC or communication protocols, but rather focus on the energy gains achieved by task shifting. In our simulations, we consider the two basic topologies depicted in Fig. 4. In both scenarios, the task of the node marked with a star symbol, is considered for being shifted to a neighboring processor node.

In our analytical evaluation, we again consider the example task of FFT computation over an input array of 1,024 bytes. Despite the fact that the output array of an FFT calculation is identical to the size of the input data, the purpose of an FFT calculation is generally the detection of particular frequency components in the signal. As opposed to the transmission of the complete resulting array, only messages with information about the input signal's amplitude at particular frequencies are returned. In order to account for this, we have defined the size of the data packet after its transmission as 50 bytes.

In scenario A, shown in Fig. 4a, a homogeneous network is considered for each of the six analyzed mote platforms. The start node transmits all data to the sink, and we consider the energy consumption of all involved nodes, i.e., the start node and all relaying nodes on the multi-hop connection to the sink. In this case, the energy demand $E_{total,A}$ of all involved nodes is $5E_{Tx} + 4E_{Rx}$, i.e., nine times the energy demand for wireless transmissions. This results from the fact that the network is assumed to be homogeneous and we have defined the energy demand for sending and receiving to be equal. The energy demand for a single transmission of the given amount of data, as well as the sum of the energy demands of all involved nodes from the source to the sink are provided in Table V.

In the second scenario, depicted in Fig. 4b, the network is assumed to be heterogeneous, and processing nodes (in this case, SunSPOT v6 devices) are depicted with triangles. In this case, the network-wide energy demand is the sum of the data transmission of the start node $E_{Tx,start}$, the reception of the data at the computing neighbor $E_{Rx,p}$, the demand for the computation $E_{compute,p}$ on the processing node, as well as the communication energy to relay the result to the sink. Based on the depicted topology, the multi-hop transmission of the results is equal to $4E_{Tx} + 3E_{Rx}$ due to the three intermediate nodes on the path to the sink. Table VI shows the energy demand of the start mote for transmitting the unprocessed data volume, the energy demand for the transmission of the processed message, and the total sum of the energy demand across all involved motes. We have again conducted this analysis for each possible starting mote type, including the SunSPOTs, which have also been selected as processor nodes.

Summarizing the results of the two considered scenarios, it can be seen that the total energy consumption E_{total} of both scenarios significantly differ. In case the processing is done within the network, and only a smaller traffic volume is forwarded to the sink, energy gains of up to 69% can be achieved when one of the computationally powerful platforms is being used for the in-network processing. Positive energy gains are ensured as tasks are only migrated when the feasibility criterion is met. The potential energy savings are remarkable, and even increase linearly with the distance to the sink because of the energy savings at each relaying node. This shows that the use of the available processing power in a heterogeneous network can reduce the overall energy demand tremendously. The proposed in-network computing can hence bring a large improvement over a homogeneous data collection approach with respect to network lifetime.

C. Network-wide Consideration

While our previous analysis has only considered a single data source, let us now analyze the energy savings for an entire data collecting WSN, as depicted in Fig. 5. As data gathering is commonly executed in a periodic fashion, we

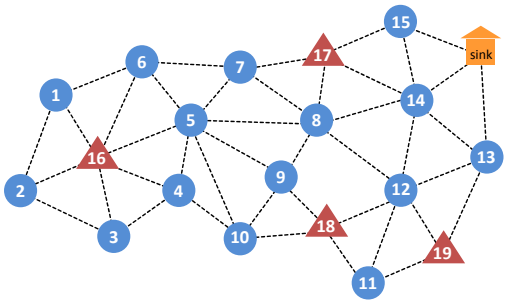


Figure 5. Topology of the sample network for which the network-wide energy savings are analyzed

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

Table VII
RESULTS OF THE CALCULATIONS FOR THE NETWORK-WIDE ANALYSIS

Mote	$E_{total,1}$	$E_{total,2}$	Savings
Imote 2	3139.135mJ	762,047mJ	76%
INGA	1233.570mJ	412.706mJ	67%
Mulle v5.2	810.575mJ	335.160mJ	59%
SunSPOT v6	3708.166mJ	866.365mJ	77%
TelosB	524.209mJ	282.661mJ	46%
XM1000	571.460mJ	291.324mJ	49%

focus on the analysis of one cycle of information gathering and the forwarding of this information to the sink. Again, we assume that each node has sampled 1,024 bytes of data and that a Fast Fourier transform needs to be applied to extract the amplitudes of relevant frequencies at a size of 50 bytes. In a first step, we assume the network of 19 data collecting nodes and one sink to be homogeneous, similar to the previous scenario A. The energy demand for all motes can then be expressed as the sum of all data transmissions. If we assume that all data packets are routed along the shortest path to the sink the total energy demand $E_{total,1}$ is equal to 133 times the energy demand for sending or receiving data.

In the second step, let us again assume the network to be heterogeneous, as depicted in the topology figure. Again, the triangles represent the processing nodes, defined as SunSPOT motes. Nodes 13, 14, and 15 transmit all their data to the sink, since they are direct neighbors. The computing nodes 16, 17, 18, and 19 perform the processing directly on their local device and only transmit the result to the sink. All 12 other motes transfer the data to a processing neighbor. The total energy demand $E_{total,2}$ is thus calculated as the sum of the energy demand for all data transmissions and all in-network processing.

Table VII shows the results of both scenarios in form of the sums of the energy demand of all motes, considering that SunSPOT devices are being used as processing nodes. The presented figures thereby equal the amount of energy consumed in each sampling cycle conducted in the WSN. The rightmost column shows the energy savings of the in-network processing as compared to the approach of transferring all collected data through a heterogeneous network without any processing. The results clearly confirm that considerable energy savings for the proposed case of in-network processing, proving the viability of our approach.

VI. CONCLUSIONS AND FUTURE WORK

Nodes with heterogeneous computational capabilities are rarely present in current WSNs. The few cases of heterogeneous mote deployments predominantly rely on two-tiered architectures, in which more powerful nodes act as distributed gateways. In the evolving Internet of Things, however, highly heterogeneous sensing and actuation platforms are expected to work in synergy. In this paper, we have shown how the resulting computational heterogeneity can be exploited in order to optimize the energy efficiency of processing data within the network.

We have presented our concept of task shifting to other nodes in the WSN, on which the actual data processing is performed. Our analysis was based on a set of six motes with different system specifications and power consumptions. After determining their real-world power consumption by means of practical high-resolution measurements, we have evaluated their energy demand in different WSN topologies. As a result of our evaluation, we have shown that the migration of processing tasks to dedicated processing nodes can extend the operational time of WSNs, even when they are only comprised of a small number of motes.

In the future, we plan to combine our approach with solutions from the domain of Mobile Agents to migrate tasks and all required input parameters between nodes, and evaluate its performance in practical settings. Furthermore, we will regard additional parameters that may have an influence on the decision to migrate a task, such as its priority or dependencies on other tasks.

ACKNOWLEDGMENT

Parts of the research leading to these results have received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 318201.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, 2002.

[2] International Telecommunication Union, *ITU Internet Reports 2005: The Internet of Things*. ITU, 2005.

[3] G. Sharma and R. Mazumdar, "Hybrid Sensor Networks: A Small World," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005, pp. 366–377.

[4] W. Hu, N. Bulusu, C. T. Chou, S. Jha, A. Taylor, and V. N. Tran, "Design and Evaluation of a Hybrid Sensor Network for Cane Toad Monitoring," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 4:1–4:28, 2009.

[5] V. Mhatre and C. Rosenberg, "Homogeneous vs. Heterogeneous Clustered Sensor Networks: A Comparative Study," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2004, pp. 3646–3651.

[6] G. Wagenknecht, M. Anwander, and T. Braun, "MARWIS: A Management Platform for Heterogeneous Wireless Sensor Networks," *ERCIM News*, no. 76, 2009.

[7] S. Rhee, D. Seetharam, and S. Liu, "Techniques for Minimizing Power Consumption in Low Data-rate Wireless Sensor Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2004, pp. 1727–1731.

Preprint of: **A. Reinhardt** and D. Burgstahler. "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes." In: *Proceedings of the 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2013, pp. 1–9.

[8] L. Yu, N. Wang, W. Zhang, and C. Zheng, "Deploying a Heterogeneous Wireless Sensor Network," in *Proceedings of IEEE Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2007, pp. 2588–2591.

[9] M. Yarvis, A. Kushalnagar, H. Singh, Y. Liu, and S. Singh, "Exploiting Heterogeneity in Sensor Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2, 2005, pp. 878–890.

[10] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A Stable Election Protocol for Clustered Heterogeneous Wireless Sensor Networks," Computer Science Department, Boston University, Tech. Rep. BUCS-TR-2004-022, 2004.

[11] T. W. Hnat, T. I. Sookoor, P. Hooimeijer, W. Weimer, and K. Whitehouse, "MacroLab: a Vector-based Macroprogramming Framework for Cyber-Physical Systems," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2008, pp. 225–238.

[12] Crossbow Technology, Inc., "Imote2 Data Sheet," available online at http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf, last access on 6 December 2012.

[13] F. Büsching, U. Kulau, and L. Wolf, "Demo Abstract: INGA - An Inexpensive Node for General Applications," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2011, pp. 435–436.

[14] Eistec AB, "Mulle Hardware Guide," available online at http://www.eistec.se/docs/Mulle_HW_Guide.pdf, last access on 6 December 2012.

[15] Oracle America, Inc., "SunSPOT data sheet," available online at http://www.sunspotworld.com/docs/Yellow/eSPOT8ds.pdf, last access on 6 December 2012.

[16] MEMSIC Inc., "TelosB Data Sheet," available online at http://memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=152:telosb, last access on 6 December 2012.

[17] Advantic Sistemas y Servicios S.L., "AS-XM1000 802.15.4 Mote Module," available online at http://www.advanticsys.com/shop/asxm1000-p-24.html, last access on 6 December 2012.

[18] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2009, pp. 1–14.

[19] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based On-line Energy Estimation for Sensor Nodes," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)*, 2007, pp. 28–32.

[20] IEEE Std, "802.15.4 Part 15.4: Wireless medium access control (MAC) and Physical layer (PHY) specifications for low-rate wireless personal are networks (LR-WPANs)," Online: http://www.ieee802.org/15/pub/TG4.html, 2006.

[21] Z. Nakutis, "Embedded Microcontrollers Benchmarking using Sliding Window Algorithm," *Electronics and Electrical Engineering – Signal Technology*, vol. 3, no. 75, pp. 53–56, 2007.

[22] M. Hempstead, M. Welsh, and D. Brooks, "Poster Abstract: TinyBench: The Case for a Standardized Benchmark Suite for TinyOS based Wireless Sensor Network Devices," in *Proceedings of the 29th Annual IEEE Conference on Local Computer Networks (LCN)*, 2004, pp. 585–586.

[23] L. Nazhandali, M. Minuth, and T. Austin, "SenseBench: Toward an Accurate Evaluation of Sensor Network Processors," in *Proceedings of the IEEE International Symposium on Workload Characterization (IISWC)*, 2005, pp. 197–203.

[24] National Climate Data Center of the USA, "Climate Data Online (CDO)," available online at http://www.ncdc.noaa.gov/cdo-web/, last access on 6 December 2012.

[25] Hitex Development Tools GmbH, "Hitex PowerScale with ACM Probe," available online at http://www.hitex.com/fileadmin/pdf/products/hardware_tools/b0-powerscale.pdf, last access on 6 December 2012.